به نام خدا

پاسخنامه آزمون «معماری نرم افزار» _ ۱۱۱۵۲۸۰

دانشجو: مریم سادات موردگر- ۹۹۰۱۹۰۴۲۶ **استاد : دکتر سید علی رضوی ابراهیمی**
رشته و مقطع تحصیلی: ارشد مهندسی نرم افزار **۱۴۰۰/۱۰/۱۹**
(پاسخ آزمون در گیت هاب به آدرس https://github.com/m-mourdgar/PNU_4001_AR نیز می باشد)

سؤالات میان ترم

۱- معماری نرم افزار چیست و نحوه ارزیابی یک معماری طرح شده را شرح دهید؟

(راهنمایی: یکی از روش های کیفی ارزیابی معماری را به زبان ساده شرح دهید.)

معماری نرم‌افزار یک برنامه یا یک سیستم محاسباتی ساختار یا ساختارهایی از سیستم است که عناصر نرم‌افزاری، خصوصیات قابل مشاهده از بیرون آن عناصر، و ارتباطات بین آنها را شامل می‌شود.

معماری نرم افزار در مورد ساختار کلی سیستم است. معماری یک سیستم اجزای اصلی آن و روابط آن ها و نحوه تعامل آن ها با یکدیگر را توصیف می کند. نقش معماری نقش اساسی در ایجاد و نگهداری نر م افزار است.

use case ها و موارد کاربردی در معماری مورد بحث هستند. در معماری صفت، کیفیت، چگونگی انجام کار، سختی یا آسانی و نیازمندی های عملیاتی (use caseها) و غیر عملیاتی و نیازمندی های سیستمی مطرح است. معماری نرم افزار، ابزار طراحی صفات کیفی است . معماری نرم افزار می تواند تاثیر زیادی بر روی صفات کیفی داشته باشد. نرم افزاری که مجهز به فضای ذخیره سازی مناسب نیست؛ اتلاف انرژی بالایی دارد. معماری نرم افزار ساختار ها و روابط آنها را توجه می کند و مجموعه ای از ساختارهایی است که لازم است در طراحی و نگهداری و ... نرم افزار مورد توجه داشته باشیم . مثلا ساختاری که با پایگاه داده نرم افزار مورد در ارتباط است، رابطه بین آن ها و خصوصیات آن ها را بحث می کند.

به زبان ساده تر می توان گفت نیازمندی ها در حالت کلی به دو دسته نیازمندی های عملیاتی و نیازمندی های غیرعملیاتی تقسیم می‌شوند. نیازمندی های عملیاتی، عبارتست از توانایی سیستم در انجام کاری که برای آن ایجاد شده است. نیازمندی های غیرعملیاتی که تحت عنوان مشخصه‌های کیفی از آنها یاد می‌شود، هر آنچه که غیر از نیازمندی های عملیاتی سیستم باشد، در این دسته قرار می‌گیرند. مانند کارایی، امنیت، هزینه ساخت و... . کیفیت نرم‌افزار به صورت مستقیم با توانایی یک سیستم در قبال نحوه انجام نیازمندی‌های عملیاتی و غیرعملیاتی آن در ارتباط می‌باشد. یک سیستم می‌تواند شامل مشخصه‌های زیادی

همچون کارایی، قابلیت نگهداری، امنیت و... باشد. کیفیت هر یک از مشخصه‌های موجود، بر کیفیت کل سیستم تاثیر دارد. یعنی کیفیت کل سیستم تابعی از کیفیت تک‌تک این مشخصه‌ها می‌باشد. لازم به ذکر است که همیشه کیفیت این مشخصه‌ها قابل اندازه‌گیری نیست.

معماری نرم افزار به ساختارهای بنیادی یک سیستم نرم افزاری و نظم ایجاد چنین ساختارها و سیستم هایی اشاره دارد. هر ساختار در نرم افزار شامل عناصر نرم افزاری، روابط بین آن ها و خصوصیات هر دو عنصر و روابط است. در واقع معماری یک سیستم، اجزای اصلی آن، روابط آن ها (ساختارها) و نحوه تعامل آن ها با یکدیگر را توصیف می کند.معماری یک سیستم نرم افزاری مشابه معماری یک ساختمان است.این معماری به عنوان یک طرح اصلی برای سیستم و پروژه در حال توسعه عمل می کند و وظایفی را که باید توسط تیم های طراحی انجام شود، مشخص می کند. معماری نرم افزار در واقع انتخاب ساختاری اساسی است که تغییر آن پس از اجرا پر هزینه است.گزینه های معماری نرم افزار شامل گزینه های ساختاری خاص از امکانات موجود در طراحی نرم افزار است.

برای ارزیابی یک معماری می شود از روش تجزیه و تحلیل تجارت معماری (ATAM) استفاده کرد که به شرح زیر می باشد :

روش تجزیه و تحلیل تجارت معماری (ATAM) بیش از یک دهه است که برای ارزیابی معماری نرم افزار در حوزه هایی از خودرو تا مالی تا دفاع ی مورد استفاده قرار می گیرد ATAM .به گونه ای طراحی شده است که ارزیابی کنندگان نیازی به آشنایی با معماری یا اهداف تجاری آن ندارند یا سیستم هنوز نیازی به ساخت ندارد و ممکن است تعداد زیادی از ذینفعان وجود داشته باشند. این روش ۹ مرحله دارد که به ترتیب زیر می باشد:

۱ - ارائه - ATAM ارائه مفهوم ATAM به ذینفعان، و پاسخ به هر گونه سوال در مورد روند کار.

۲- ارائه محرک های تجاری - همه افراد در این فرآیند، محرک های تجاری سیستم مورد نظر را ارائه و ارزیابی می کنند.

۳- ارائه معماری - معمار با "سطح مناسبی از جزئیات" معماری سطح بالا را به تیم ارائه می دهد.

۴ - شناسایی رویکرد های معماری - روش های مختلف معماری برای سیستم توسط تیم ارائه شده و در مورد آنها بحث می شود.

۵ - تولید درخت سودمندی صفت کیفی - در این مرحله الزامات اصلی تجاری و فن ی سیستم را تعریف می شود و آنها را با ویژگی مناسب معماری ترسیم می کنند. سناریویی نیز برای این نیاز ارائه می شود.

۶- تحلیل رویکردهای معمار ی - تجزیه و تحلیل هر سناریو ، رتبه بندی آنها بر اساس اولویت و سپس ارزیابی معماری در برابر هر سناریو.

۷- طوفان فکری و اولویت بندی سناریو ها - در میان گروه ذ ینفعان بزرگتری سناریوهای موجود ارائه می شود و گسترش پیدا می کند.

۸ - تجزیه و تحلیل رویکردهای معماری – مرحله ۶ با اطلاعات اضافه شده گروه بزرگتر ذینفعان دوباره انجام می شود.

۹ - ارائه نتایج – کلیه اسناد و مدارک در اختیار ذینفعان قرار می گیرد.

این مراحل در دو فاز از هم تفکیک می شوند: فاز ۱ شامل مراحل ۱ تا ۶ است و پس از این فاز ، وضعیت و زمینه پروژه ، نیازهای معماری محرک و وضعیت اسناد معماری مشخص می شود. فاز ۲ که شامل مراحل ۷ تا ۹ است و ارزیابی در آن به پایان می رسد.

سؤالات پایان ترم

۱- مدل یک صفت کیفی را بطور کامل شرح دهید.

قابلیت دسترسی Availability

خصوصیت کیفی قابلیت دسترسی در ارتباط با شکست سیستم و پیامدهای همراه با آن است. شکست سیستم زمانی رخ می دهد که سیستم سرویس های پیش فرض خود را به طور مناسب ارائه نکند. همچنین وقتی شکست روی می دهد نتایج آن توسط کاربران سیستم قابل مشاهده است. حوزه های مرتبط با این خصوصیت کیفی عبارتند از :

- چگونه شکست سیستم تشخیص داده می شود.
- چگونه شکست سیستم رخ می دهد.
- زمانی که یک شکست رخ می دهد چه اتفاقی می افتد.
- تا چه مدت سیستم در حالت عملیاتی نیست (در صورت شکست).
- چگونه میتوان جلوی شکست را گرفت.
- زمانی که یک شکست رخ می دهد چه نوع اخطارهای لازم است.

باید توجه کرد که میان دو مفهوم شکست و نقص تفاوت وجود دارد. یک نقص در صورت اینکه مورد بررسی قرار نگیرد و برطرف نشود منجر به شکست می شود. یک شکست توسط کاربر سیستم قابل مشاهده است در حالی که نقص توسط کاربر قابل مشاهده نیست. زمانی که یک نقص توسط کاربر قابل مشاهده می شود در واقع آن تبدیل به شکست شده است. به عنوان مثال، یک خرابی می تواند انتخاب الگوریتم اشتباه برای یک محاسبه باشد که نهایتا نتیجه اشتباه حاصل شده توسط الگوریتم اشتباه منجر به شکست می شود.

یکی از مهمترین مفاهیم در زمانیکه سیستم دچار شکست می شود مدت زمان لازم برای تعمیر است. از آنجایی که شکست سیستم توسط کاربران قابل مشاهده است بنابراین زمان برطرف کردن شکست برابر است با زمان بین مشاهده شدن شکست توسط کاربر تا زمانی که دیگر شکست قابل مشاهده نیست (سیستم تعمیر می شود).

تفاوت بین شکست و نقص ارائه کننده مفهوم راهبردهای تعمیر خودکار است. معنی راهبرد تعمیر خودکار اینکه در زمان اجرای سیستم اگر نقصی روی دهد، سیستم بدون آنکه شکست را نشان دهد (دچار شکست شود) آن نقص را اصلاح کند. بنابراین با این راهبرد، دیگر در سیستم شکست وجود نخواهد داشت.

در دسترس بودن یک سیستم برابر است با احتمال آنکه سیستم در زمان مورد نیاز قابل استفاده باشد.

سناریوهای عمومی قابلیت دسترسی

منبع تحریک: بین تحریک های داخلی و خارجی مشخص کننده شکست و نقص، تفاوت وجود دارد زیرا پاسخ سیستم به هر یک از این تحری کها میتواند متفاوت باشد.

محرک: یک نقص که میتواند محرک در نظر گرفته شود در یکی از دسته های زیر قرار می گیرد:

Omission : یک مولفه شکست خورده که نمی تواند به ورودی پاسخ دهد.

Crash : مولفه مکررا دچار نقص از نوع omission می شود.

Timing : یک مولفه پاسخ میدهد ولی پاسخ زود یا دیر تولید می شود.

Response : یک مولفه پاسخ می دهد ولی پاسخ آن صحیح نیست.

فرآورده: فرآورده مشخص کننده منبعی است (مانند پردازشگر، کانال ارتباطی، فرآیند یا انباره) که نیازدارد تا حد بالایی قابل دسترس باشد.

محیط: وضعیت سیستم وقتی نقص یا شکست اتفاق می افتد ممکن است پاسخ سیستم را تحت تاثیر قرار دهد. به عنوان مثال اگر سیستم از قبل دچار چندین نقص شده و در حال اجرا در یک حالت غیرطبیعی است احتمالا بهتر است که سیستم را کاملا خاموش کرد اما اگر اولین نقص مشاهده شده باشد می توان بعضی ملاحظات را در مورد زمان پاسخ یا نحوه عملکرد سیستم، در نظر گرفت.

پاسخ: برای یک شکست واکنش های متفاوتی می تواند وجود داشته باشد. این واکنش ها می نتوا ند ثبت کردن شکست، هشدار دادن به کاربر یا دیگر سیستم ها، خاموش کردن سیستم های خارجی و یا دسترس پذیر نبودن در طول تعمیر در نظر گرفته شوند.

معیار پاسخ: معیار پاسخ می تواند درصد قابلیت دسترسی، مدت زمان تعمیر و زمان ها یا دوره هایی که سیستم در آنها قابل دسترس است، در نظر گرفته شود.

۲- یک صفت کیفی تخیلی را طراحی و جدول سناریوهای عمومی آن و یک نمونه سناریوی معین برای آن تعریف نمائید. (مثلا: «دوست داشتنی» یا «دانشجویی!» یا هر صفت کیفی تخیلی و ابتکاری دیگر)

صفت کیفی مُسِن پسند

در دنیای امروز که برای انجام بسیاری از کارهای روزمره نیاز به استفاده از اپلیکیشن ها و نرم افزارهای مختلف است، لازم است برخی از نرم افزارها به گونه ای باشند که افراد مُسِن و کم سواد جامعه نیز بتوانند از آن ها استفاده کنند. مانند نرم افزارهای بانک یا تهیه بلیط و یا بیمه ها و ...

این صفت می خواهد که یک نرم افزار خیلی ساده و واضح باشد تا همه افراد بتوانند از آن استفاده کنند. از منوهای کمتر استفاده شود در نتیجه کارهای قابل اجرای کمتری نیز باید در برنامه گنجانده شود و نرم افزار کارهای متفاوتی را انجام ندهد.

مثلا می توان از نرم افزار موجود بانک ها استفاده کرد و آن ها را برای این هدف بهینه کرد تا در اختیار کاربران این گروه قرار بگیرد.

سناریوهای عمومی مُسِن پسند:

منبع: کاربر نهایی (افراد مُسِن و کم سواد)

محرک: یادگیری ویژگی های سیستم – احساس راحتی هنگام کار با سیستم

فرآورده: نرم افزار مورد استفاده

محیط: رابط کاربری

پاسخ: حمایت ازیادگیری ویژگی های سیستم مثل واسط آشنا به کاربر و واسط کاربری که در یک زمینه ناآشنا قابل استفاده است. حمایت از استفاده کارا از سیستم مثل استفاده مجدد از داده ها و فرمان های از قبل وارد شده و پرهیز از فیلدهای اضافه و گیج کننده، استفاده از خط واضح و خوانا برای نوشته ها و گرفتن تایید نهایی برای انجام عملیات

معیار پاسخ: سرعت بارگذاری، سرعت واکنش به درخواست، رضایت کاربر مُسِن

مثال نرم افزار بانک عملیات انتقال موجودی

سناریوی عمومی

کاربر مُسِن

صفت مُسِن پسند

قابلیت ذخیره شماره کارت کاربر

قابلیت ذخیره شماره کارت مقصد

قابلیت جایگذاری خودکار رمز دوم پس از درخواست آن

در صورت وارد نکردن شماره کارت مقصد یا اشتباه بودن آن خطا می دهد.

سناریوی معین:

حالت خاص: اگر رمز اشتباه وارد شود بدون پاک شدن اطلاعات وارد شده دیگر تا سه بار اجازه ورود مجدد رمز را بدهد.

کاربر مُسِن

رمز اشتباه است و بدون پاک کردن دیگر فیلدهای وارد شده اجازه ورود مجدد رمز را می دهد.